

AD-A128 651

STATE-OF-THE-ART ASSESSMENT OF TESTING AND TESTABILITY  
OF CUSTOM LSI/VLSI..(U) AEROSPACE CORP EL SEGUNDO CA  
ENGINEERING GROUP M A BREUER ET AL. OCT 82

1/1

UNCLASSIFIED

TR-0083(3902-04)-1-VOL-3 SD-TR-83-20-VOL-3 F/G 14/2

NL

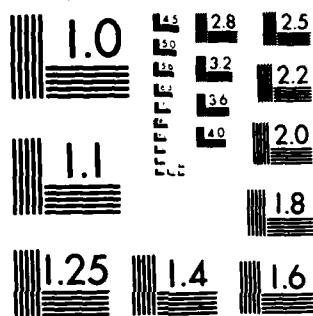
END

DATE

FILMED

6 83

DTIC



MICROCOPY RESOLUTION TEST CHART  
NATIONAL BUREAU OF STANDARDS-1963-A

12

# State-of-the-Art Assessment of Testing and Testability of Custom LSI/VLSI Circuits

Volume III: Fault Mode Analysis

M. A. BREUER & ASSOCIATES  
Encino, Calif. 91436

and

A. J. CARLAN  
Technical Study Director

October 1982

Engineering Group  
THE AEROSPACE CORPORATION  
El Segundo, Calif. 90245

Prepared for  
SPACE DIVISION  
AIR FORCE SYSTEMS COMMAND  
Los Angeles Air Force Station  
P.O. Box 92960, Worldway Postal Center  
Los Angeles, Calif. 90009

DTIC  
ELECTE  
MAY 26 1983  
S D  
B

APPROVED FOR PUBLIC RELEASE:  
DISTRIBUTION UNLIMITED

83 05 26 104

AD A128651

DTIC FILE COPY


This final report was submitted by the Aerospace Corporation, El Segundo, CA 90245 under Contract No. F04701-82-C-0083 with the Space Division, Deputy for Logistics and Acquisitions, P.O. Box 92960, Worldway Postal Center, Los Angeles, CA 90009. It was reviewed and approved for The Aerospace Corporation by J. R. Coge, Electronics and Optics Division, Engineering Group. Al Carlan was the project engineer.

This report has been reviewed by the Office of Information and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

FOR THE COMMANDER

APPROVED

  
STEPHEN A. HUNTER, LT COL, USAF  
Director, Speciality Engineering  
and Test

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER SD-TR-83-20	2. GOVT ACCESSION NO. <b>A128651</b>	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) State-of-the-Art Assessment of Testing and Testability of Custom LSI/VLSI Circuits Vol III: Fault Mode Analysis		5. TYPE OF REPORT & PERIOD COVERED Interim
7. AUTHOR(s) M.A. Breuer & Associates and A.J. Carlan, Aerospace Technical Director		6. PERFORMING ORG. REPORT NUMBER TR-0083(3902-04)-1
9. PERFORMING ORGANIZATION NAME AND ADDRESS M.A. Breuer & Associates 16857 Bosque Dr. Encino, CA 91436		8. CONTRACT OR GRANT NUMBER(s) F04701-80-C-0081 F04701-81-C-0082 F04701-82-C-0083
11. CONTROLLING OFFICE NAME AND ADDRESS Space Division Air Force Systems Command Los Angeles, Calif. 90245		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) The Aerospace Corporation El Segundo, Calif. 90245		12. REPORT DATE October 1982
		13. NUMBER OF PAGES 44
		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Fault Modes Fault Modeling Single Stuck Line Faults Multiple Stuck Line Faults Heuristic Testing Methods <i>Complete copy retained in the document file</i>		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Physical failure in LSI/VLSI circuits is highly dependent on the fabrication technology being used and result in a very complex faulty behavior. To reduce numbers and types of faults that must be handled for test generating and fault simulation, logic fault models are used. The most popular fault model is the single stuck line (SSL) which can emulate many common physical faults. Non-standard faults like short circuits are more difficult to model-usually require modification to the original circuit to allow use of SSL software. This approach is also ideal for handling CMOS faults.		

DD FORM 1473  
(FACSIMILE)

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

## EXECUTIVE SUMMARY

A large number of specific physical fault modes have been recognized to occur in digital systems due to manufacturing defects and various wearout mechanisms. These failures are usually highly dependent on the fabrication technology being used, and may result in very complex faulty behavior. To reduce the numbers and types of faults that must be handled during test generation and fault simulation to manageable levels, various logical fault modes have been proposed, in which failures are characterized by their effects on the logical structure and behavior of the system under consideration. The use of logical rather than physical fault models simplifies fault analysis, and makes it relatively independent of circuit technology. However, not all fault modes that occur in practice can be easily or accurately modeled in this manner. For example, some lines and components appearing in a physical circuit have no counterparts in the corresponding logic circuit and vice versa.

Logical fault modes can be classified in terms of their time-variance, the number of primitive faults present simultaneously, and the fault's effect on component behavior, interconnection structure, and operating speed. By far the most widely used fault model is the single stuck line or SSL model. An SSL fault allows any single signal line in a circuit to be permanently stuck at the logic value 0 or 1; component behavior and operating speed are unaffected. The popularity of the SSL model has several reasons. Many common physical faults are

PRECEDING PAGE BLANK-NOT FILMED

83 05 26 104

equivalent to SSL faults. The line-by-line analysis characteristic of test generation techniques like the D-algorithm makes SSL faults very easy to handle. Finally, practical experience indicates that test sets derived for SSL faults thoroughly exercise a circuit, thereby detecting many faults that cannot be modeled directly as SSL faults. However, except in simple cases, it is very difficult to identify the non-SSL faults covered by a given test set for SSL faults. Thus to guarantee 100 percent fault coverage, it is generally necessary to consider other fault modes in addition to SSL faults.

If several signal lines are allowed to be stuck simultaneously, then the multiple stuck line or MSL fault model is obtained. MSL faults are difficult to deal with directly, because their number grows exponentially with the number of lines present. In practice, a complete set of SSL tests can be expected to cover all MSL faults. An MSL fault can escape detection only if certain complex masking conditions are present. Short-circuit faults are more difficult to deal with. Their number is also large and, unlike stuck-line faults, they can introduce unwanted feedback. The occurrence of short-circuit and other non-standard fault modes can be minimized by careful circuit layout.

Non-standard faults like short circuits are usually modeled by modifying the original circuit so that an SSL fault can be introduced that is equivalent to the target fault in the unmodified circuit. Although such "workarounds" are costly to construct, they allow standard SSL-based test software to be applied to most nonstandard faults. This approach can be used, for example, to handle CMOS faults that introduce

"parasitic" memory elements. Another fault mode found in MOS VLSI circuits is pattern sensitivity caused by unwanted signal interactions. Promising fault models for pattern sensitive faults in random-access memories have been devised, but they have not been incorporated into test generation software. Heuristic testing methods, whose underlying fault modes are not explicitly defined, continue to be very widely used for complex faults.



Accession For	
NTIS GRA&I	<input checked="checked" type="checkbox"/>
DTIC TAB	<input type="checkbox"/>
Unannounced	<input type="checkbox"/>
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	



## TABLE OF CONTENTS

	page
EXECUTIVE SUMMARY .....	iii
1. INTRODUCTION .....	1
2. PHYSICAL VS. LOGICAL FAULTS .....	3
IC Faults .....	3
Logical Fault Models .....	5
3. STANDARD FAULT MODEL .....	13
Model Definition .....	13
Advantages .....	14
Disadvantages .....	17
4. OTHER FAULT MODELS .....	19
Multiple Stuck-line Faults .....	19
Short Circuit Faults .....	21
Pattern-Sensitive Faults .....	25
Miscellaneous Faults .....	29
5. BIBLIOGRAPHY .....	35

PRECEDING PAGE BLANK-NOT FILMED

## 1. INTRODUCTION

This report examines the fault modes that have been used or proposed as a basis for test generation in digital systems, particularly systems employing very large scale integration (VLSI). Of main concern are logical fault models which can be specified in terms of changes in the logical properties of the unit under test. Also of interest are the relationships among the various fault modes which allow tests generated for one type of fault to be used to detect other types of faults.

Chapter 2 is concerned with the relationship between the physical fault modes encountered in practical circuits, and the logical fault models that have been proposed to model them. The main physical fault modes of integrated circuits are summarized, including wiring faults, metallization and dielectric faults, parametric faults, and soft failures. The use of logical models to replace physical faults is discussed. It is shown that this simplifies fault analysis and makes it relatively independent of the particular circuit technology being used. A set of criteria for classifying fault modes is presented.

The remaining two chapters examine specific fault modes in detail. The most widely used fault model, the single stuck line or SSL model, which allows any one line in a circuit to be stuck at logical 0 or 1, is the subject of Chapter 3. The characteristics of this model are described, and its advantages and disadvantages are analyzed. It is noted that while there are many faults that are not equivalent to SSL faults

they can often be detected by tests generated for SSL faults. Representative non-standard fault types are considered in Chapter 4, including multiple stuck line or MSL faults, short circuit faults, pattern-sensitive faults, soft faults, and certain faults peculiar to CMOS circuits. The ability of SSL tests to detect these non-standard faults is also discussed. The report concludes with a bibliography on faults in digital circuits.

## 2. PHYSICAL VS. LOGICAL FAULTS

Extensive studies have been made of the physical failure modes occurring in integrated circuits [Schnable and Keen 1971, Case 1976, Edwards 1980, Partridge 1980]. Figure 2.1 summarizes the most common IC fault modes. They have two major sources: defects in the manufacturing process, and component wearout. The frequency of occurrence and relative importance of the various faults depends on the circuit type (TTL, ECL, NMOS, CMOS, etc.) and the manufacturing technology used.

### IC Faults

Package wiring faults are typically caused by failure in the connections between IC pads and the IC package pins. For example, the metal-to-metal bond formed between a pad and a connecting wire can fail forming an open circuit. Another class of wiring faults are due to failures of the on-chip metal connects, which are usually made from aluminum (Al). Contaminants like moisture inside the IC package may cause corrosion in metal connectors. A phenomenon called electromigration, which is the tendency of Al atoms to flow in the direction of an applied electric current, can cause Al connectors to thin out and break. Breaks can also occur due to microcracks appearing at steps in an Al connector that is deposited over discontinuities in the underlying dielectric (usually  $\text{SiO}_2$ ). Short-circuit faults can be caused by metal bridging the space between two adjacent metal connectors, a result for example of inadequate etching of the space between the connectors.

- Package wiring faults
- On-chip metallization (aluminum) faults due to:
  - Corrosion
  - Electromigration
  - Microcracks
  - Bridging
- Dielectric (silicon dioxide) faults due to:
  - Mask defects
  - Electrostatic discharge
- Surface faults
- Threshold shifts
- Pattern sensitivity
- Soft faults due to:
  - Alpha particles
  - Cosmic rays

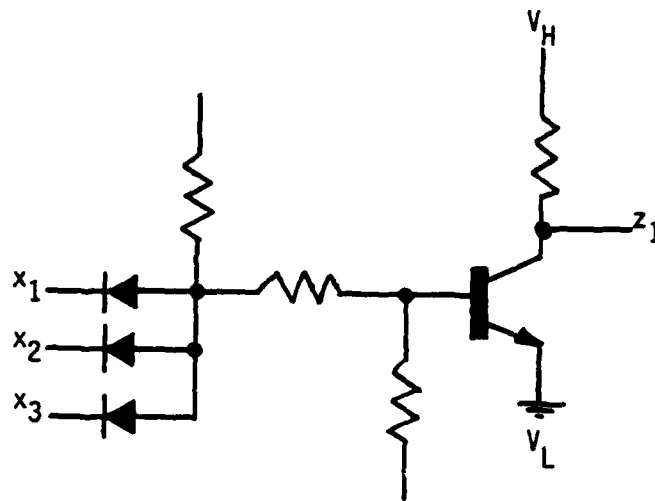
Fig. 2.1. Representative physical failure modes in integrated circuits.

An IC may also fail due to physical defects in the  $\text{SiO}_2$  dielectric used to insulate the various devices on the chip. Defects like dust particles on the photographic masks used during manufacture can create holes (pinhole defects) in an  $\text{SiO}_2$  layer. Such holes can also be caused by electrostatic discharge due to improper handling or shielding. The latter problem is most common in MOS circuits which contain very thin oxide layers.

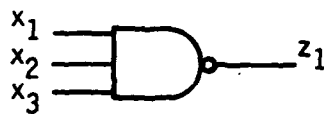
Other physical fault modes listed in Fig. 2.1 include changes in the IC's surface state resulting in excessive leakage currents. Threshold voltages in MOS circuits can shift due to the movement of electric charges on or in an  $\text{SiO}_2$  layer. Unwanted interactions between signals that are adjacent in time or space can result in pattern sensitivity. Pattern sensitive faults are a consequence of the high component densities characteristic of VLSI, and are usually found in chips containing large random-access memories (RAMs). Also common in high-density RAMs are the so-called "soft" failures caused by radiation that can effectively alter a logical state represented by a stored charge. Trace amounts of radioactive elements in the IC package material produce  $\alpha$ -particles that can cause intermittent errors by destroying a charge packet. Similar effects can be caused by cosmic radiation.

#### Logical Fault Models

Given any physical fault mechanism in a digital circuit, it is always possible, at least in principle, to determine its effect on the logical behavior of the circuit. For example, Fig. 2.2a shows a



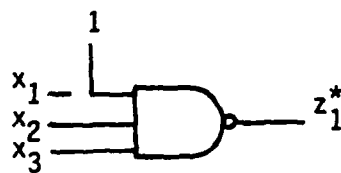
(a)



(b)

$x_1$	$x_2$	$x_3$	$z_1$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(c)



(d)

$x_1$	$x_2$	$x_3$	$z_1^*$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

(e)

Fig. 2.2. (a-c) A diode-transistor NAND gate  $G$ . (d-e) The effect of the fault " $x_1$  shorted to  $V_H$ " on  $G$ .

diode-transistor realization of a 3-input NAND gate  $G$ , which is represented symbolically in Fig. 2.2b. Two voltage levels  $V_H$  (high) and  $V_L$  (low) define the logic values 1 and 0, respectively. The logical behavior of  $G$  is described by the truth table of Fig. 2.2c. Suppose that the following physical fault  $f_p$  occurs in  $G$ : input line  $x_1$  is accidentally connected (short circuited) to  $V_H$ . This could be expected to have the effect shown in Fig. 2.2d where the faulty output function  $z_1^*$  is now independent of the value of  $x_1$ . We can readily define a logical fault  $f_L$  which mirrors precisely the effect of  $f_p$ . Disconnect input line  $x_1$  from the fault-free gate  $G$ , and apply a constant logical 1 to the disconnected input as shown in Fig. 2.2e. This logical fault, usually referred to as "line  $x_1$  stuck at logical 1" is equivalent to  $f_p$ . Note that while  $f_p$  may be an internal fault in  $G$ , the logical model assumes that  $G$  is fault-free and associates a fault with an interconnection line.

There are several advantages to using logical instead of physical fault models in digital fault mode analysis.

(1) Once we have a logical fault model that adequately reflects the physical failure modes of a circuit, fault analysis becomes a logical rather than a physical problem.

(2) It is possible to construct logical fault models that are applicable to many different technologies, in which case fault analysis becomes relatively technology-independent. This means that computer programs for fault simulation and test generation can be written that do not lose their usefulness with changes in technology. For example, although the diode-transistor circuit technology illustrated in



Fig. 2.2a is now largely obsolete, the "stuck-at-0/1" logical fault model used in this example is applicable to essentially all types of semiconductor circuits.

(3) Using logical fault models it may be possible to derive tests for faults whose physical cause is unknown, or whose effect on circuit behavior is not completely understood.

(4) A logical fault model often covers a large number of different physical faults, resulting in a substantial decrease in the complexity of fault analysis.

Various criteria may be used for classifying both physical and logical faults; Fig. 2.3 lists the most important ones. It is usually assumed that faults are permanent, i.e., invariant with respect to time. Time-varying faults do occur, particularly intermittent faults, where the circuit moves in an apparently random fashion between the fault-free state and some fixed faulty state. An intermittent fault can only be detected if an appropriate test pattern is applied to the circuit under test during one of its faulty periods. The models that have been proposed for analyzing intermittent faults [Ball and Hardy 1959, Breuer 1973] require statistical data on the probability of fault occurrence; unfortunately this type of information is normally unavailable. Transient faults also occur randomly, but, unlike intermittent faults, are not repetitive; soft errors induced by  $\alpha$ -particles are usually considered to be transient.

- Variability with respect to time:
  - Permanent
  - Intermittent
  - Transient
- Number of primitive faults that may be present simultaneously:
  - Single faults
  - Multiple faults
- Effect on components
- Effect on interconnections between components
- Effect on operating speed

Fig. 2.3. Parameters for classifying faults in digital systems.

Another basic assumption in fault mode analysis is that only single faults affecting just one component or connection need be detected. This assumption is justified in field testing by the fact that the probability of a single fault is usually much greater than that of a multiple fault. Thus if testing is carried out often enough, most of the faults that are encountered will be single faults. Moreover, most multiple faults can be detected by testing for the individual single faults of which they are composed. It is difficult to deal with multiple faults directly because their number tends to be extremely large. If a particular (logical) fault model allows  $n$  independent single faults, then there are  $\binom{n}{k}$  possible multiple faults of multiplicity  $k$ . For example, if  $n=1000$ , then there are about half a million double faults ( $k=2$ ) and about  $1.66 \times 10^8$  triple faults ( $k=3$ ). For a newly manufactured part, the probability of a multiple fault is usually much higher than that of a single fault. Again it is usually assumed that a comprehensive test set based upon the single fault assumption will detect faulty components containing multiple faults.

The types of faults encountered in circuit components can often be restricted by very general physical arguments. Consider, for example, the 3-input NAND gate  $G$  of Fig. 2.2 which realizes the function  $z_1 = \overline{x_1 x_2 x_3}$ . It is unlikely that a fault can change  $z_1$  to say  $\overline{x_1 + x_2 + x_3}$  (the NOR function), because this would require a major change in the switching threshold of the gate, i.e., in net input signal level at which the output signal changes value. Similarly, a change to the EXCLUSIVE-OR function  $x_1 \oplus x_2 \oplus x_3$  is highly improbable as it would require an increase in the number of threshold levels. It can be assumed for most technol-

ologies that a fault will not change the basic gate type; a faulty NAND gate remains a NAND. This assumption enables us to restrict the possible faulty function for  $G$  to the following set of eight NAND functions:  $0, 1, \overline{x_1 x_2}, \overline{x_1 x_3}, \overline{x_2 x_3}, \overline{x_1}, \overline{x_2}, \overline{x_3}$ . This is a small subset of the 256 distinct switching functions of up to three variables.

Faults that change circuit interconnection topology include open circuits (the breaking of a connection) and short circuits (the establishment of a connection between two normally unconnected points). Open circuit faults are fairly easy to model. The signal source end of an open connection has no further effect on the circuit. The sink end of the connection generally remains fixed at the logical 0 or 1 value. Short circuit behavior is much more difficult to analyze, and fault models tend to be technology dependent.

Changes in the signal propagation delay of the various components and connections in a logic circuit can also cause erroneous behavior. For example, suppose the input pattern  $(x_1, x_2, x_3)$  applied to the 3-input NAND  $G$  is required to change at time  $t$  from  $(1, 1, 0)$  to  $(1, 0, 1)$ . This should cause no change in the value of  $z_1$  which should remain at 1. Suppose, however, that due to changes in signal delays,  $x_3$  changes value  $\Delta t$  seconds before  $x_2$ , then  $(1, 1, 1)$  will be applied to  $G$  for  $\Delta t$  seconds. If  $\Delta t$  is sufficiently long to overcome the inertia inherent in every physical device, a spurious 0 signal will appear at  $z_1$ . The propagation delays of logic gates and connections are difficult to measure and can vary within wide tolerances. For this reason, conservative or worst-case design techniques are usually employed, as well as synchronization (clock) signals to compensate for minor variations in

signal delays. Direct analysis of delay faults is feasible only in certain types of circuits [Lesser and Shedletsky 1980].

### 3. STANDARD FAULT MODEL

Most work to date in digital system testing, both theoretical and applied, has employed a logical fault model that is referred to here as the *single stuck line* or SSL model. In this chapter the SSL model is defined, and its advantages and disadvantages are analyzed.

#### Model Definition

In a single stuck line (SSL) fault, exactly one logical connection in a circuit may be permanently stuck at logical one (s-a-1) or stuck at logical zero (s-a-0). Circuit components like gates and flip-flops are assumed to be fault-free. Figure 2.2 illustrates a typical SSL fault, in this case a primary input line of a NAND gate that is s-a-1. In terms of the fault classification of Fig. 2.3, SSL faults have the following attributes:

- Faults are permanent or time-invariant.
- Only one fault is present at a time.
- Circuit components are unaffected by the fault.
- Any logic line may be stuck at 0 or 1.
- Operating speed is unaffected.

In order to analyze the usefulness of the SSL fault model (or any other model), the concepts of fault equivalence and fault dominance are helpful. Two physical or logical faults  $F_1$  and  $F_2$  in a given circuit  $N$  are *equivalent*, denoted  $F_1 = F_2$ , if  $N$  realizes the same circuit

function with  $F_1$  present as it does with  $F_2$  present. In other words,  $F_1$  and  $F_2$  have exactly the same effect on the behavior of  $N$ . A test  $T$  detects  $F_1$  if and only if it detects  $F_2$ .  $F_1$  and  $F_2$  can be distinguished by external testing only if  $F_1 \neq F_2$ . If every test  $T$  that detects  $F_1$  also detects  $F_2$ , we say that  $F_1$  *dominates*  $F_2$ . Fault dominance is a weaker relationship than fault equivalence, since  $F_1$  can dominate  $F_2$  without being equivalent to  $F_2$ .

### Advantages

There are several reasons why the SSL model has become the classical logical fault model for digital fault analysis.

(1) Many physical faults, such as open-circuited lines, and lines that are short-circuited to power or ground, are equivalent to SSL faults. They can therefore be precisely analyzed with this logical model.

(2) Test generation for SSL faults is relatively easy. Most test generation methods, such as the D-algorithm and LASAR [Breuer and Friedman 1976]\* use a technique called path sensitization illustrated by Fig. 3.1. Suppose that the SSL fault (output of)  $G_0$  s-a-1 is to be detected. It suffices to apply a pattern of input signals to the circuit which creates a path (shown by the heavy line in Fig. 3.1) over which an error signal can propagate from the faulty line to an observable

---

\* See also Report No. AC 1.81 "Test Generation" in this series.

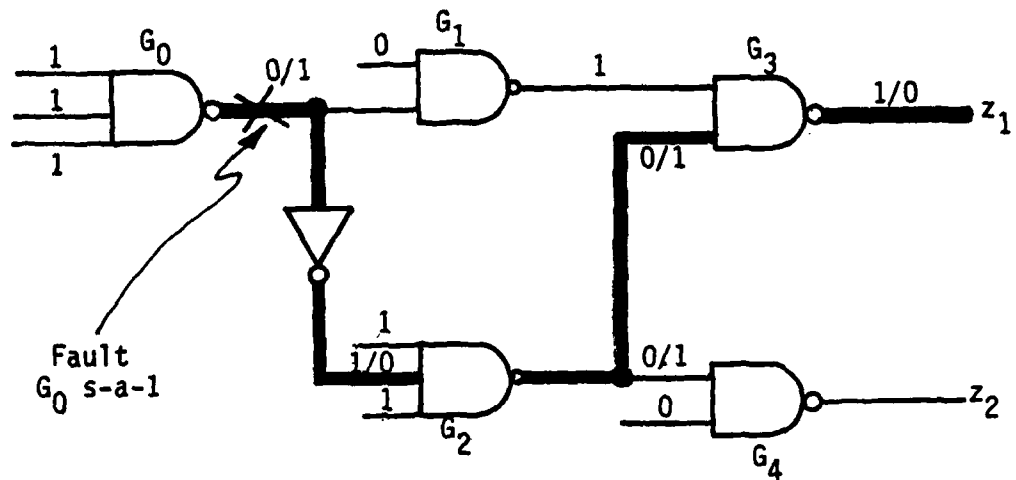


Fig. 3.1. Detection of an SSL fault via a sensitized path (heavy line).



output of the circuit, in this case the output line  $z_1$ . This path is said to be sensitized since any change in the signal applied to the input end of the path propagates to the output end. The fact that an SSL fault is associated with a single line in the circuit makes it well suited to the line-by-line analysis required for path sensitization.

(3) Using standard path sensitization techniques, tests for many SSL faults can be generated simultaneously thereby reducing test generation costs. Consider again the circuit of Fig. 3.1 where it is desired to obtain a test for the SSL fault  $G_0$  s-a-1. Let  $E/\bar{E}$  denote the signal on a line in cases where the state of the line is  $E$  if no fault is present and  $\bar{E}$  if  $G_0$  is s-a-1. Thus a test for this fault must apply 0/1 to the output line of  $G_0$ . It must also apply a signal of the form  $E/\bar{E}$  to every line along the sensitized path in order for the desired error-signal propagation to take place. This implies that the given test pattern will detect an SSL fault of the s-a- $\bar{E}$  type associated with every line on the sensitized path. If the test generation procedure creates long sensitized paths, then a test for many SSL faults will be constructed in one step. This feature of the SSL model is exploited in the widely-used LASAR test generation program [Thomas 1971].

(4) The number of SSL faults that must be dealt with is relatively small and manageable. For example, if a circuit contains  $N$  distinct logic lines, then there are  $2N$  possible SSL faults. This number can be further reduced by using equivalence and dominance properties to eliminate faults that will be detected by tests generated for other faults.

(5) Experience accumulated over the past twenty years suggests that many fault types that are not equivalent to SSL faults, such as shorts between logic signals and multiple stuck-line faults, are nevertheless detected by test sets constructed to detect SSL faults. In other words, many fault modes are dominated by SSL faults. This may be explained intuitively by the fact that a complete set of tests for SSL faults exercises every line in a circuit and propagates the effect of this exercising to the primary outputs. Furthermore, a reasonably complete set of test patterns is applied to each gate in the circuit thus activating internal non-SSL-equivalent fault modes. This follows from the fact that at least  $n+1$  test patterns must be applied to any  $n$ -input logic gate of the AND, OR, NAND, NOR or NOT type in order to detect the SSL faults associated with the gate's input and output lines. The average number of inputs per gate  $n_{av}$  in a typical digital system is around 2 or 3. For  $n_{av} = 3$ , a set of SSL based tests must therefore apply at least four test patterns to the average gate, which is 50% or more of the possible input patterns that exist for the gate. This can be regarded as exercising the average gate fairly thoroughly for most possible faults.

#### Disadvantages

In spite of the above advantages, the SSL model does not encompass all testing problems. There are many physical lines in integrated circuits, for example, power and ground lines, which do not appear in standard logic diagrams; conversely not all lines in logic diagrams correspond to physical connectors [Galiay et al. 1980]. Although an SSL-

based test set can be expected to cover most faults occurring in practical circuits, to guarantee 100 percent coverage of such faults, non-SSL faults must be explicitly considered. Furthermore, there are many fault situations where one or more of the basic assumptions underlying the SSL model does not hold. For instance, soft faults (see Fig. 2.1), unlike SSL faults, are transient, and so are likely to escape detection by SSL tests that are applied periodically under the assumption that a fault remains after it first occurs. The single-fault assumption may not be valid for faults occurring during manufacture, which frequently affect many lines and components of a circuit. Multiple faults are also encountered during field testing, especially when maintenance intervals are long, thereby allowing single faults to accumulate. The analysis of such non-SSL faults is considered in the following chapter.

#### 4. OTHER FAULT MODELS

There are numerous situations where known physical faults are not equivalent to or dominated by SSL faults. To handle such faults, a variety of logical models have been proposed. The more important of these nonstandard fault modes are examined in this chapter.

##### Multiple Stuck-Line Faults

A straightforward way to extend the standard SSL model of the preceding chapter is to lift the constraint that only one logical connector may be s-a-0 or s-a-1 at any time. The result is the *multiple stuck line* or MSL fault model. As in the SSL model, faults are assumed to be permanent, and the circuit components and circuit operating speed are assumed to be fault-free. MSL faults are of interest for several reasons.

- (1) They can be used to model far more physical faults than the SSL model. If a circuit contains  $N$  distinct logic lines, there are  $2N$  possible SSL faults, but there are  $3^N - 1$  possible MSL faults.
- (2) As IC component density increases, physical failures are more likely to affect many lines and components. Such multiple faults more closely resemble MSL than SSL faults.
- (3) Use of the MSL model can sometimes simplify SSL fault analysis. For example, in generating SSL tests for all-NAND circuits, s-a-0 faults can be ignored if one is willing to consider both single and multiple s-a-1 faults [Hayes 1971]. In a sequential circuit, an SSL fault may,

over a period of time, propagate faulty signals to many different parts of the circuit. Multiple fault models may then be most appropriate for analyzing the subsequent behavior of the circuit [Breuer and Friedman 1976].

The main difficulty in dealing with MSL faults is the vast number of such faults, which grows exponentially with the number of lines present. Hence no commercial test generation systems deal explicitly with MSL faults. However, it has long been recognized that the SSL faults tend to dominate the MSL faults. In other words, a set of tests  $T$  that detect all SSL faults in a circuit is likely to detect most, if not all, MSL faults. This is to be expected in view of the fact that  $T$  must detect each component SSL fault  $f_i$  of a multiple fault  $F = (f_1, f_2, \dots, f_k)$  comprising  $k$  stuck lines. It is possible, however, for  $T$  to fail to detect some MSL fault due to the phenomenon of *fault masking*. Suppose, for example, that in addition to the target fault  $f_1 = G_0$  s-a-1, the circuit of Fig. 3.1 contains a second fault  $f_2 = G_1$  s-a-1. The given test pattern detects  $f_1$ , but does not detect the MSL fault  $(f_1, f_2)$ . This is due to the fact that error propagation from  $f_1$  to  $z_1$  along the sensitized path is blocked or masked at  $G_3$  by the presence of  $f_2$ .

In general, an SSL test can fail to detect a multiple fault  $(f_1, f_2, \dots, f_k)$  if the detection of each component fault  $f_i$  is masked by the remaining faults. The masking conditions for undetectability are relatively complex, and not frequently encountered in practice. This justifies the common assumption that SSL tests are adequate for MSL detection. However, it is usually not possible to guarantee that a given SSL test set covers all MSL faults without detailed analysis. A few classes of combinational logic are known for which complete sets of SSL

tests are guaranteed to detect all MSL faults. Schertz has defined a *restricted logic circuit* [Schertz and Metze 1972] as one that satisfies the following two conditions.

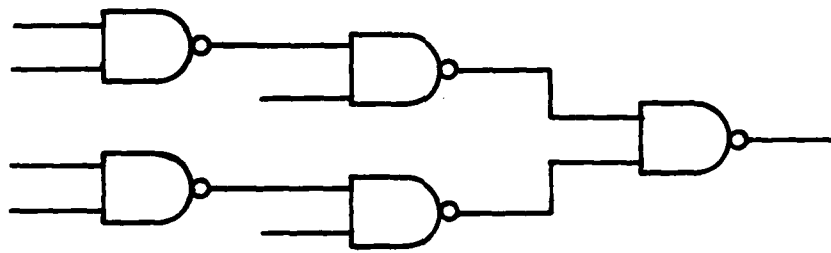
(1) It contains no subcircuit with the same interconnection pattern as the 5-gate circuit of Fig. 4.1a.

(2) Only the primary input lines may fan out to two or more gates.

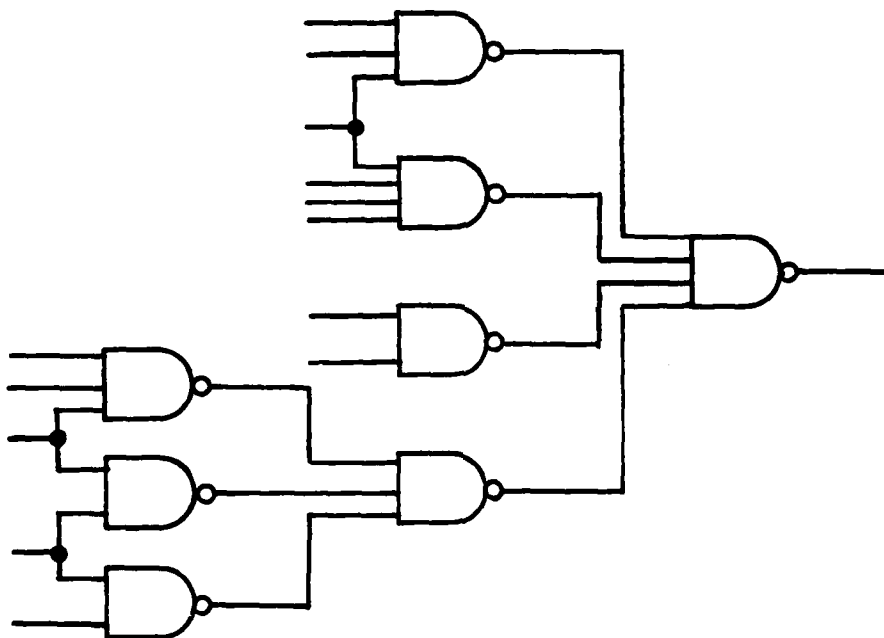
Figure 4.1b shows such a circuit. Restricted circuits include the important class of 2-level circuits. They have the very useful property that every complete SSL test set is also a complete MSL test set. A slightly weaker result exists for the class of fanout-free circuits. A *fanout-free circuit* is one in which every signal line is connected to at most one gate. This implies that there is only one path from every line to the primary output. It can be shown [Hayes 1971] that for every fanout-free circuit there exists a minimal SSL test set which is also an MSL test set. Not every SSL test set for a fanout-free circuit detects all multiple faults, but simple procedures are known for generating minimal SSL test sets that also detect all MSL faults.

#### Short Circuit Faults

A large percentage of the physical faults occurring in modern integrated circuits can be classified as short-circuit (SC) or bridging faults, where two normally disconnected logic lines become connected [Nicholls 1979]. Short circuit faults are significantly more



(a)



(b)

Fig. 4.1. (a) A minimal unrestricted combinational circuit.  
(b) Example of a restricted circuit.

complex than stuck-line faults, since the signals associated with the shorted lines are, in general, variable rather than constant. Furthermore, the number of single SC faults, i.e., faults involving just one pair of shorted lines, in an  $N$ -line circuit is  $\binom{N}{2} = N(N-1)/2$ , which is considerably more than the  $2N$  possible SSL faults. Based upon the layout of a circuit it is possible to determine pairs of wires between which a short is impossible [Galiay et al. 1981]. Based upon such an analysis the total number of short faults which need be considered can be drastically reduced.

An SC fault, unlike an SSL or MSL fault, can introduce spurious feedback that converts a combinational circuit into a sequential circuit. Short circuits can also result in unexpected fault modes that depend on the circuit technology being used. For example, in diode-transistor logic circuits like that of Fig. 2.2a, a shorted diode in an input line of a gate  $G_1$  can affect the output of a different gate  $G_2$ , if  $G_1$  and  $G_2$  derive inputs from a common source [Friedman 1974].

An SC fault involving two signal lines  $z_1$  and  $z_2$  has the general form depicted in Fig. 4.2. In many current circuit technologies the new logic signals  $\varphi_1(z_1, z_2)$  and  $\varphi_2(z_2, z_3)$  created by the fault correspond to wired logic functions where  $\varphi_1 = \varphi_2 = \varphi$ , and either  $\varphi(z_1, z_2) = z_1 z_2$  (wired-AND case) or  $\varphi(z_1, z_2) = z_1 + z_2$  (wired-OR case). However, in a significant number of cases  $\varphi$  assumes an indeterminate value different from 0 or 1. Research into SC fault modes has generally been restricted to faults that result in well-defined wired-AND or wired-OR behavior [Mei 1974, Karpovsky and Su 1980].



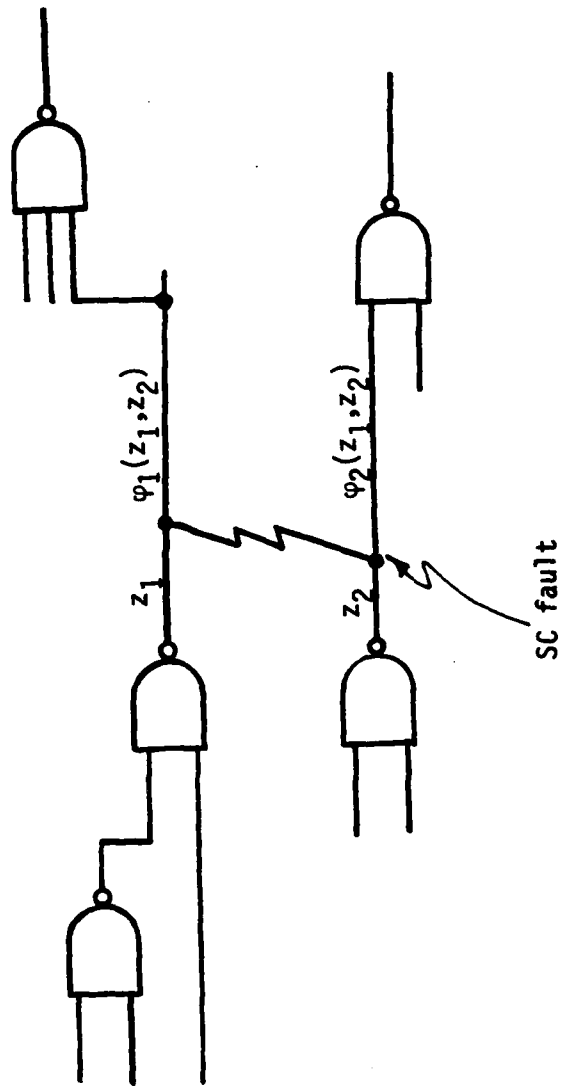


Fig. 4.2. Typical short circuit fault.

As in the case of MSL faults experience indicates that most SC faults are detected by a complete set of SSL tests. For certain SC faults this can be guaranteed. An *input bridging fault* is an SC fault that only involves the input lines of a single gate. It is assumed that there is no point of fanout between the site of the short circuit and the gate in question. It can be shown that every SSL test set detects all input bridging faults of this type [Friedman 1974, Mei 1974].

A general technique for modeling SC faults has been proposed in [Kaposi and Kaposi 1972]. Suppose that a short circuit between lines  $z_1$  and  $z_2$  as in Fig. 4.2 results in a wired OR. For fault analysis, replace these two lines by the circuit shown in Fig. 4.3. Under fault-free conditions  $\varphi_1(z_1, z_2) = z_1$  and  $\varphi_2(z_1, z_2) = z_2$ . If the fault  $f = \text{line } x \text{ s-a-1}$  is present in this circuit, then both  $\varphi_1$  and  $\varphi_2$  change to  $z_1 + z_2$ . Hence the problem of detecting the SC fault in the original circuit has been transformed into the problem of detecting the SSL fault  $f$  in the modified circuit. This modeling technique can be extended to a wide class of faults, its usefulness being limited by the complexity of the modified circuit.

#### Pattern Sensitive Faults

VLSI, with its great increase in the component density of integrated circuits, has stimulated interest in a new class of faults called pattern-sensitive faults (PSFs) or adjacent pattern interference faults (APIs) [Hayes 1975 and 1980, Srini, 1977, Nickel 1980]. PSFs are caused by unwanted interference between signals that are adjacent in time or space. They are most prevalent in high-density ICs like RAMs where com-

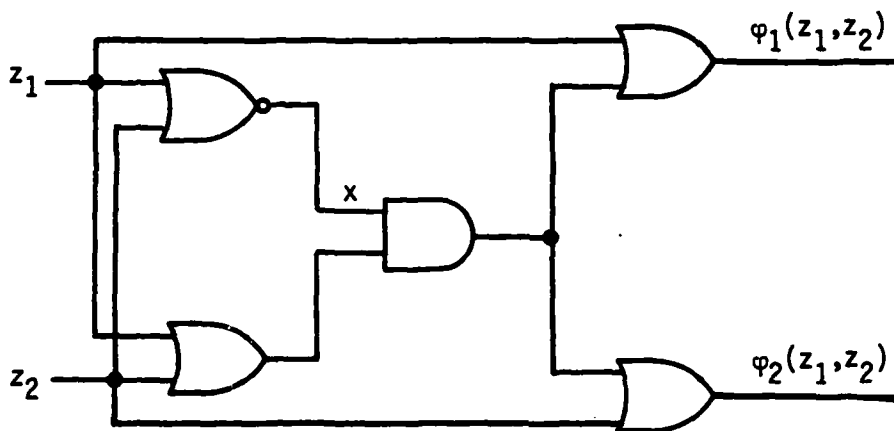


Fig. 4.3. Model for a wired-OR short circuit fault.

ponents (memory cells and signal lines) are packed very closely together, and signal-to-noise ratios are relatively small. PSFs result primarily from electromagnetic interaction between closely-packed lines and signals. They are particularly difficult to detect because they are often both data-pattern and data-rate dependent. Consequently they may appear to be intermittent or transient faults.

A variety of different physical fault mechanisms are responsible for PSFs, making it difficult to define logical fault models to represent them. Several such models have been proposed for PSFs in RAMS. A very general model is described in [Hayes 1975] in which each 1-bit cell  $C_i$  of a RAM is associated with a set of cells  $N(C_i)$  called the neighborhood of  $C_i$ .  $N(C_i)$  defines the range of spatial pattern sensitivity around  $C_i$ . A PSF is said to be present in  $C_i$  if a read or write operation addressed to  $C_i$  affects or is affected by the data pattern stored in  $N(C_i)$ . This is a type of functional fault model which requires detection of essentially any functional change in the sub-RAM defined by  $N(C_i)$ . The problem of generating tests for this kind of fault is equivalent to the classical problem of generating a checking sequence for a sequential machine [Friedman and Menon 1971]. It is not difficult to generate checking sequences for RAMs because of their relatively simple state behavior. However, the length of these sequences increases exponentially with neighborhood size. Hence in order to obtain test sequences of manageable size, it is necessary to restrict  $N(C_i)$  to a small number of cells.

Several studies of RAM PSFs have been carried out in which  $N(C_i)$  is restricted to  $C_i$  and its four or six nearest neighbors [Srini 1977, Hayes 1980, Suk and Reddy 1980]. In general, if  $N(C_i)$  forms a  $k$ -cell neighborhood of regular structure, a so-called tiling neighborhood, then all PSFs in an  $n$ -bit RAM can be detected by a test sequence comprising at most  $(3k+2)2^k n$  reads and writes [Hayes 1980]. For example, if  $k=5$ , then the test length is  $544n$ , a manageable number. Test length can be further reduced by imposing additional restrictions on PSF behavior. Suk and Reddy, for example, have developed efficient test generation methods under the assumption that all reads and certain write operations cannot induce PSFs [Reddy & Suk 1979, Suk & Reddy 1980]. Nair et al. also define a simplified PSF model using the concept of cell coupling [Nair et al. 1978]. Two cells are coupled if a write operation addressed to one cell of a pair causes the other cell to change state. Like the Suk and Reddy model, coupled-cell faults include PSFs affecting read operations only. Test generation methods are known for detecting all faults of this kind involving coupled cell-pairs; however, these methods have not been extended to coupled faults involving three or more cells. In general, the various PSF models discussed above have not been verified experimentally, so their validity remains open to question. In practice, PSF detection methods are based on heuristic procedures like the GALPAT test [Breuer and Friedman 1976], which employ no explicit fault model.

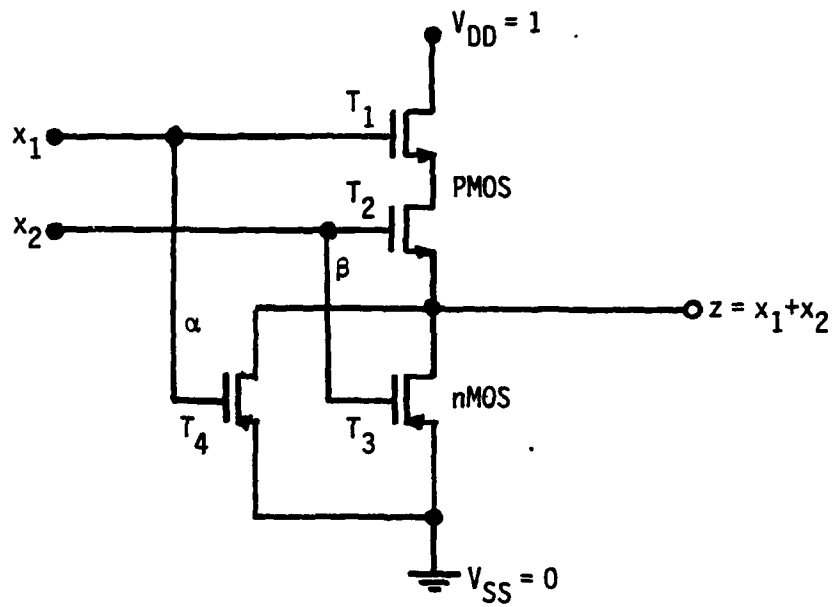
### Miscellaneous Faults

The faults discussed in the preceding sections are time-invariant or "hard" faults. The increasing chip densities of VLSI circuits, particularly RAMs, has resulted in an increase in the frequency of transient or "soft" faults caused primarily by alpha particle radiation [May 1979, May et al. 1980]. These soft faults are of two main types: *storage faults* that cause the state of a memory element to change (from 0 to 1 or from 1 to 0), and *sensing faults* that cause incorrect logic signals to appear on address or sense amplifier lines. In each case, no permanent damage is done to the circuit. Since such soft errors are by their nature random and non-recurring, the only practical ways to deal with them are by shielding the chip to prevent the radiation from reaching it, by using circuit design and layout techniques that minimize sensitivity to the radiation, or by using error-correcting circuits to correct the erroneous logic signals produced by the soft faults after they occur.

MOS VLSI circuits exhibit a number of nonstandard fault modes. These circuits often employ a third logic value, the high-impedance state Z, permitting a connector to become stuck-at-Z (s-a-Z) or stuck-at open [Wadsack 1978]. Anomalous behavior can also result from the fact that logic signals are represented by charge packages that are stored in the gate capacitance of an MOS transistor. If this charge packet becomes isolated due to one or more s-a-Z faults, then the signal in question no longer responds correctly to external signals, and the corresponding transistor acts as a spurious or "parasitic" flip-flop [Sievers and Avizienis 1981].

Figure 4.4 illustrates the foregoing failure modes for the case of a 2-input CMOS NOR gate that realizes the logical function  $z = \overline{x_1 + x_2}$ . When the input pattern  $(x_1, x_2) = (1, 0)$  is applied to this gate, the transistors  $T_1$  and  $T_3$  are turned off, while  $T_2$  and  $T_4$  are turned on. Consequently, a path exists from the primary output to  $V_{SS}$  (ground) via  $T_4$  making  $z = 0$  when no faults are present. Now suppose that the nMOS transistor  $T_4$  is missing, or that its gate input line is open-circuited. This fault mode corresponds to the presence of the stuck-line fault  $\alpha$  stuck-at-Z, denoted by  $\alpha/Z$ . When  $(x_1, x_2) = (1, 0)$  with the fault  $\alpha/Z$  present,  $T_4$  is not turned on, causing line  $z$  to become isolated from all sources of input signals. As indicated in Fig. 4.4b,  $z$  retains its previous value  $X$  in the form of the stored charge associated with the output devices connected to  $z$ . If  $X \neq 0$ , then an incorrect value is perceived at  $z$  which remains present until either the associated charge at  $z$  leaks away, or else the input signals applied to the gate are changed. The other stuck-at-Z faults listed in Fig. 4.4b behave similarly.

Just as short-circuit faults can be modeled by equivalent s-a-0/1 faults in the manner illustrated in Fig. 4.3, the non-classical faults of Fig. 4.4 can also be modeled by a standard logic circuit with s-a-0/1 faults. Figure 4.5 shows a logical model that can be used for fault analysis in this CMOS NOR gate [Wadsack 1978]. It consists of a D flip-flop of unit delay and several zero-delay gates  $G_1:G_4$ . The D flip-flop represents the capacitance associated with the output line  $z$ . When  $T = 1$ , this flip-flop stores the previous state  $X$ . Under normal conditions  $T = 1$ ,  $z(t) = X(t-1)$ , and the circuit behaves like a NOR



(a)

Input		Output z			
$x_1$	$x_2$	Normal	$\alpha/Z$	$\beta/Z$	$V_{DD}/Z$
0	0	1	1	1	X
0	1	0	0	X	0
1	0	0	X	0	0
1	1	0	0	0	0

(b)

Fig. 4.4. (a) A 2-input CMOS NOR gate

(b) Effect of some non-classical fault modes; X denotes previous state.



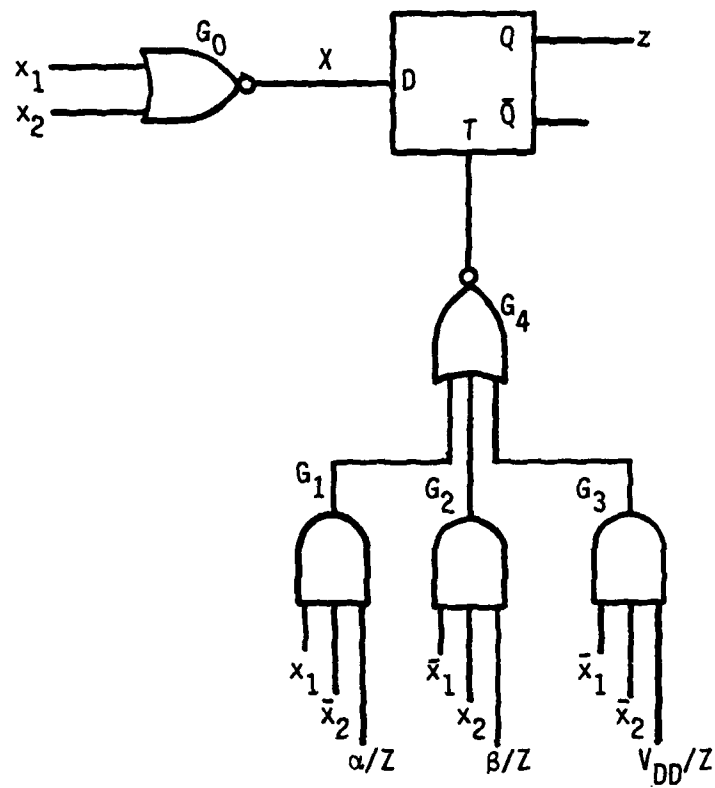


Fig. 4.5. Logic model for the NOR gate and non-classical faults of Fig. 4.4.

gate with unit delay. Each of the three s-a-Z faults of Fig. 4.4b is modeled by a s-a-1 fault on an input to one of the AND gates  $G_1:G_3$ . For example, to model the effect of  $\alpha/Z$ , the corresponding input line of  $G_1$  is set to 1. When the sensitizing input combination  $(x_1, x_2) = (1, 0)$  occurs, the output of  $G_1$  becomes 1 forcing T to D, thereby latching the previous state X into the D flip-flop.

Another class of circuits whose special fault modes have been investigated are PLAs (programmable logic arrays) [Ostapko and Hong 1979, Smith 1979]. PLAs are subject to faults occurring at array crosspoints where a desired connection is missing, or an unwanted connection is present; such crosspoint faults are closely related to SSL faults [Smith 1979].

The fault models discussed so far are only applicable to systems for which detailed, usually gate-level, circuits are available. Relatively little attention has been devoted to the development of higher-level fault models of a kind suitable for register-level circuits in which low-level faults like SSL faults cannot be seen. Of particular interest are functional fault models in which fault modes can be directly related to the logical operation or function of the circuit under consideration [Breuer and Friedman 1980, Sridhar and Hayes 1981]. It is clear that more research attention must be devoted to higher-level fault modes in the future if the testing problems associated with VSLI-based systems are to be satisfactorily addressed.

## 5. BIBLIOGRAPHY

- [Ball and Hardy 1969] H. Ball and F. Hardy, "Effects and detection of intermittent failures in digital systems," *Proc. AFIPS Conf. (1969 FJCC)*, vol. 35, pp. 329-335, 1969.
- [Breuer 1973] M.A. Breuer, "Testing for intermittent faults in digital circuits," *IEEE Trans. Computers*, vol. C-22, pp. 241-246, March 1973.
- [Breuer and Friedman 1976] M.A. Breuer and A.D. Friedman, *Diagnosis and Reliable Design of Digital Systems*, Woodland Hills, California, Computer Science Press, 1976.
- [Breuer and Friedman 1980] M.A. Breuer and A.D. Friedman, "Functional level primitives in test generation," *IEEE Trans. Computers*, vol. C-29, pp. 223-235, March 1980.
- [Case 1976] G.R. Case, "Analysis of actual fault mechanisms in CMOS logic gates," *Proc. 13th Design Automation Conf.*, pp. 265-270, San Francisco, June 1976,
- [Edwards 1980] D.G. Edwards, "Testing for MOS integrated circuit failure modes," *Digest 1980 Test Conf.*, Philadelphia, pp. 407-416, November 1980.
- [Friedman 1974] A.D. Friedman, "Diagnosis of short-circuit faults in combinational logic circuits," *IEEE Trans. Computers*, vol. C-23, pp. 746-752, July 1974.
- [Friedman and Menon 1971] *Fault Detection in Digital Circuits*, Prentice-Hall, Englewood Cliffs, New Jersey, 1971.
- [Galiay et al. 1980] J. Galiay, Y. Crouzet, and M. Vergnault, "Physical versus logical fault models in MOS LSI circuits: impact on their testability," *IEEE Trans. Computers*, vol. C-29, pp. 527-531, June 1980.
- [Hayes 1971] J.P. Hayes, "A NAND model for fault diagnosis in combinational logic networks," *IEEE Trans. Computers*, vol. C-20, pp. 1496-1506, December 1971.
- [Hayes 1975] J.P. Hayes, "Detection of pattern sensitive faults in random access memories," *IEEE Trans. Computers*, vol. C-24, pp. 150-157, February 1975.
- [Hayes 1977] J.P. Hayes, "Modeling faults in digital logic circuits," *Rational Fault Analysis* (ed. R. Saeks and S.R. Liberty), New York, Marcel Dekker, pp. 78-95, 1977.

PRECEDING PAGE BLANK-NOT FILMED

- [Hayes 1980] J.P. Hayes, "Testing memories for single-cell pattern-sensitive faults," *IEEE Trans. Computers*, vol. C-29, pp. 249-254, March 1980.
- [Kaposi and Kaposi 1971] J.F. Kaposi and A.A. Kaposi, "Testing switching networks for short circuit faults," *Electronic Letters*, vol. 8, pp. 586-587, November 1971.
- [Karpovsky and Su 1980] M. Karpovsky and S.Y.H. Su, "Detection and location of input and feedback bridging faults in combinational networks," *IEEE Trans. Computers*, vol. C-29, pp. 523-527, June 1980.
- [Lesser and Shedletsky 1980] J.D. Lesser and J.J. Shedletsky, "An experimental delay test generator for LSI logic," *IEEE Trans. Computers*, vol. C-29, pp. 235-248, March 1980.
- [May 1979] T.C. May, "Soft errors in VLSI — present and future," *IEEE Trans. Components, Hybrids and Manufacturing Technology*, vol. CHMT-2, no. 4, pp. 377-387, December 1979.
- [May et al. 1980] T.C. May et al., "Soft error testing," *Digest 1980 Test Conf.*, Philadelphia, pp. 137-150, November 1980.
- [McAteer 1979] O.J. McAteer, "Shocking blow to military electronics," *Mil. Electronics/Countermeasures*, pp. 59-63, June 1979.
- [Mead and Conway 1980] C. Mead and L. Conway, *Introduction to VLSI Systems*, Reading, Massachusetts, Addison-Wesley, 1980.
- [Mei 1974] K.C.Y. Mei, "Bridging and stuck-at faults," *IEEE Trans. Computers*, vol. C-23, pp. 720-727, July 1974.
- [Nair et al. 1978] R. Nair, S.M. Thatte and J.A. Abraham, "Efficient algorithms for testing semiconductor random-access memory," *IEEE Trans. Computers*, vol. C-27, pp. 572-576, June 1978.
- [Nicholls 1979] D.B. Nicholls, "Digital evaluation and generic failure analysis data," Reliability Analysis Center, Rome Air Development Center, Griffis AFB, New York, Report MDR-10, January 1979.
- [Nickel 1980] V.V. Nickel, "VLSI — the inadequacy of the stuck at fault model," *Proc. GOMAC 80*, pp. 331-334, 1980.
- [Ostapko and Hong 1979] D.L. Ostapko and S.J. Hong, "Fault analysis and test generation for programmable logic arrays (PLA)," *IEEE Trans. Computers*, vol. C-28, pp. 617-627, September 1979.
- [Partridge 1980] J. Partridge, "Testing for bipolar integrated circuit failure modes," *Digest 1980 Test Conf.*, Philadelphia, pp. 397-406, November 1980.

- [Reddy and Suk 1979] S.M. Reddy and D.S. Suk, "Test procedures for semiconductor random access memories," Rome Air Development Center, Tech. Rept. RADC-TR-79-269, November 1979.
- [Schertz and Metze 1972] D.R. Schertz and G. Metze, "A new representation of faults in combinational digital circuits," *IEEE Trans. Computers*, vol. C-21, pp. 858-866, August 1972.
- [Schnable et al. 1978] G.L. Schnable, T.J. Gallace, and H.L. Pryor, "Reliability of CMOS integrated circuits," *Computer*, vol. 11, no. 10, pp. 6-17, October 1978.
- [Schnable and Keen 1971] G.L. Schnable and R.S. Keen, "On failure mechanisms in large-scale integrated circuits," *Advances in Electronics and Electron Phys.*, vol. 30, pp. 79-138, 1971.
- [Sievers and Avizienis 1981] M.W. Sievers and A. Avizienis, "Analysis of a class of totally self-checking functions implemented in an MOS LSI general logic structure," *Digest 11th Symp. on Fault Tolerant Computing*, pp. 256-261, June 1981.
- [Smith 1979] J.E. Smith, "Detection of faults in programmable logic arrays," *IEEE Trans. Computers*, vol. C-28, pp. 845-853, November 1979.
- [Sridhar and Hayes 1981] T. Sridhar and J.P. Hayes, "A functional approach to testing bit-sliced microprocessors," *IEEE Trans. Computers*, vol. C-30, pp. 563-571, August 1981.
- [Srini 1977] V.P. Srini, "API tests for RAM chips," *Computer*, vol. 10, no. 7, pp. 32-35, July 1977.
- [Suk and Reddy 1980] D.S. Suk and S.M. Reddy, "Test procedures for a class of pattern-sensitive faults in random-access memories," *IEEE Trans. Computers*, vol. C-29, pp. 419-429, June 1980.
- [Thomas 1971] J.J. Thomas, "Automated diagnostic test programs for digital networks," *Computer Design*, pp. 63-67, August 1971.
- [Wadsack 1978] R.L. Wadsack, "Fault modeling and logic simulation of CMOS and MOS integrated circuits," *BSTJ*, vol. 57, pp. 1449-1474, May-June 1978.